

BAB 2

TINJAUAN PUSTAKA DAN DASAR TEORI

1.1 Tinjauan Pustaka

Berikut data perbandingan penelitian yang dilakukan dengan penelitian lain yang sudah dibuat sebelumnya. Dapat dilihat pada tabel 2.1

Tabel 2.1 Data Perbandingan Penelitian

Parameter Penulis	Objek	Metode	Library AR	Bahasa Pemrograman	Platform
Mohammad Syahrofi Irsyad (2016)	Simulasi Ikatan Kimia	Fast Corner Detection	Vuforia	C#	Android
Riani Indriani, Bayu Sugiarto, Agus Purwanto (2016)	Pengenalan Hewan	Image Tracking	Vuforia	C#	Android
Dekhi Aryanto (2014)	Visualisasi Tata Surya	Marker Based Tracking	OpenSpace3d	ActionScript	Desktop
Galan Januanesbi (2014)	Pembelajaran Vulkanologi	Marker Based Tracking	ARToolkit	Java	Desktop
Apri Santoso, Elki Noviandi (2013)	Pembelajaran Organ Tubuh	Marker Based Tracking	ARToolkit	Java	Desktop
Diusulkan : (2017)	Pengenalan Percampuran Warna	Distance Marker	Vuforia	C#	Android

Penelitian menggunakan teknologi *augmented reality* pernah Mohammad Syahrofi Irsyad, 2016. Metode yang digunakan adalah *fast corner detection* dan dibuat menggunakan *library* Vuforia. Aplikasi yang dihasilkan menampilkan objek 3D atom sesuai dengan *marker* dan terdapat *marker bond* yang jika didekatkan dengan *marker* atom akan terjadi ikatan atom. Aplikasi ini dijalankan pada *platform* android.

Penelitian dengan topik sejenis juga dilakukan oleh Riani Indriani, Bayu Sugiarto dan Agus Purwanto, 2016. Pada penelitian yang dilakukan digunakan metode *image tracking* dan dibuat menggunakan *library* Vuforia. Aplikasi yang dihasilkan menampilkan objek 3D hewan sesuai dengan *image target* yang dideteksi dilengkapi dengan animasi serta informasi mengenai hewan tersebut. Aplikasi ini dijalankan pada *platform* android.

Penelitian yang lain juga dilakukan oleh Dekhi Aryanto, 2014. Metode yang digunakan pada penelitian ini digunakan metode *marker based tracking* dan dibuat dengan *library* OpenSpace3d. Aplikasi yang dihasilkan menampilkan objek 3D planet pada layar monitor saat *marker* dikenali. Aplikasi ini dijalankan pada *platform desktop*.

Penelitian tentang *augmented reality* dengan *marker target* juga dilakukan oleh Galan Januanesbi, 2014. Dalam penelitian ini digunakan metode *marker based tracking* dan dibuat dengan *library* ARToolkit. Aplikasi yang dihasilkan menampilkan objek 3D gunung berapi disertai dengan animasi pada layar monitor saat *marker* dikenali. Aplikasi ini dijalankan pada *platform desktop*.

Penelitian lain juga dilakukan oleh dilakukan Apri Santoso dan Elki Noviandi, 2013. Pada penelitian ini metode yang digunakan adalah *marker based tracking* dan dibuat dengan *library* ARToolkit. Aplikasi yang dihasilkan menampilkan objek 3D organ tubuh manusia pada layar monitor saat *marker* dikenali. Aplikasi ini dijalankan pada *platform desktop*.

Dari penelitian sebelumnya *augmented reality* diaplikasikan sebagai media pengenalan atau simulasi dan digunakan *marker/image* sebagai target untuk objek 3D. Adapun yang membedakan dengan penelitian sebelumnya adalah *augmented reality* diaplikasikan untuk media pengenalan percampuran warna pada *platform* android dan percampuran warna ini terjadi melalui interaksi antar *marker*.

2.2 Dasar Teori

2.2.1 *Augmented Reality*

Augmented Reality (AR) adalah penggabungan benda-benda nyata dan maya di lingkungan nyata, berjalan secara interaktif dalam waktu nyata (*real-time*), dan biasanya terdapat integrasi antar benda dalam tiga dimensi, yaitu benda maya terintegrasi dalam dunia nyata. Lebih lanjut AR merupakan variasi dari lingkungan *virtual* atau yang lebih dikenal dengan istilah *virtual reality* (VR). Teknologi VR membuat pengguna seperti berada di dalam sebuah lingkungan secara keseluruhan. Saat berada di lingkungan *virtual* tersebut, pengguna tidak bisa melihat lingkungan nyata di sekitarnya. Sebaliknya, teknologi AR memungkinkan pengguna untuk melihat lingkungan nyata, dengan objek *virtual* yang ditambahkan atau digabungkan dengan lingkungan nyata. (Ronald T. Azuma, 1997).

Menurut Stephen Cawood dan Mark Fiala, AR merupakan cara alami untuk mengeksplorasi objek 3D dan data, AR merupakan suatu konsep perpaduan antara *virtual reality* dan *world reality*. Sehingga objek-objek virtual 2D dan 3D seolah-olah terlihat nyata dan menyatu dengan dunia nyata. Pada teknologi AR, pengguna dapat melihat dunia nyata yang ada di sekelilingnya dengan penambahan objek virtual yang dihasilkan oleh komputer (Cawood, 2008).

2.2.2 Warna

Warna adalah suatu hasil dari sifat cahaya yang terpancar dan ditangkap oleh mata sebagai indera penglihat. Warna dapat didefinisikan secara obyektif/fisik sebagai sifat cahaya yang dipancarkan, atau secara subyektif/psikologis sebagai bagian dari pengalaman indera penglihatan. Secara obyektif atau fisik, warna dapat diberikan oleh panjang gelombang. Dilihat dari panjang gelombang, cahaya yang tampak oleh mata merupakan salah satu bentuk pancaran energi yang merupakan bagian yang sempit dari gelombang elektromagnetik. (Sadjiman Ebdi Sanyoto, 2005).

Menurut kejadiannya, warna dibagi menjadi dua, yaitu warna *additive* dan *subtractive*. Warna *additive* adalah warna yang berasal dari cahaya biasa disebut spektrum. Warna pokok *additive* adalah merah (red), hijau (green) dan biru (blue) yang dalam komputer disebut model warna RGB. (Wirania Swasty, 2010).

Nilai intensitas dari warna cahanya adalah antara 0 - 255. Nilai 0 untuk hitam dan 255 untuk intensitas maksimum dari merah, hijau biru atau RGB. (RoseGISLabs, 2016)

Intensitas Merah	Intensitas Hijau	Intensitas Biru	Warna
255	255	0	Kuning
0	255	255	Cyan
255	0	255	Magenta
127	0	0	Mid – red
127	127	127	Mid – grey
0	0	0	Hitam
255	255	255	Putih
241	0	171	Ungu
255	155	50	Orange

Gambar 2.1 Contoh Nilai Intensitas Kombinasi Warna RGB
(RoseGISLabs, 2016)

2.2.3 Vuforia

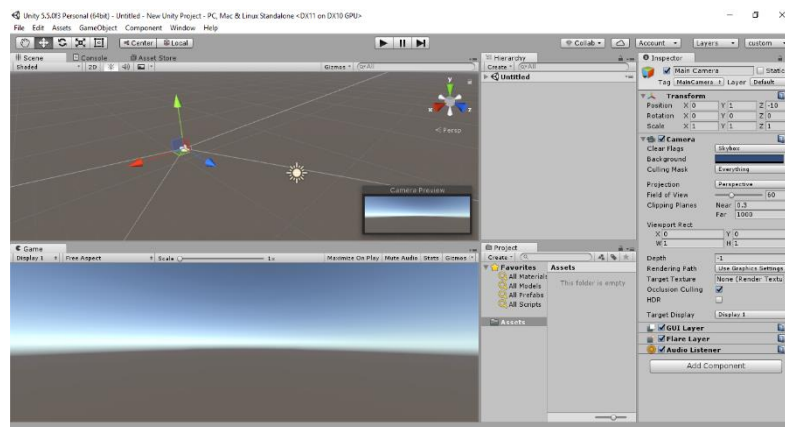
Vuforia adalah (*Software Development Kit*) SDK yang dikembangkan oleh Qualcomm yang digunakan sebagai pendukung pembuatan aplikasi-aplikasi *augmented reality* (AR) pada *platform mobile* (android, ios). Vuforia menganalisa gambar dengan mendeteksi *marker* dan akan menampilkan informasi 3D dari *marker* yang sudah dideteksi tersebut.

Vuforia mendukung berbagai macam jenis *image target* 2D dan 3D *marker* termasuk *image target markerless*. Fitur lain dari vuforia yaitu deteksi oklusi lokal menggunakan tombol *virtual*. Vuforia menyediakan API (*aplication programing interface*) pada lingkungan C++, java dan objective C. (Vuforia Developer, 2017).

2.2.4 Unity 3D

Unity 3D adalah sebuah tool game engine yang digunakan untuk mempermudah proses pembuatan game dan aplikasi 2D, 3D, VR dan AR. Game engine ini dilengkapi dengan GUI (graphic user interface) yang memudahkan untuk

membuat, mengedit serta membuat script untuk menciptakan sebuah game 3D. Unity 3D mendukung multi platform, game yang dibuat menggunakan game engine ini dapat di build ke banyak platform device diantaranya PC (windows, mac, linux), android, ios, windows phone, tizen, web, game console (PS, xbox, nitendo wii) bahkan sampai smart tv. Unity 3d mendukung 3 bahasa pemrograman dalam fitur scriptingnya yaitu JavaScript, C# dan Boo. (Unity3D, 2017).



Gambar 2.2 User Interface Unity 3D

2.2.5 Marker

Marker merupakan sebuah penanda khusus yang memiliki pola tertentu sehingga saat kamera mendeteksi *marker*, objek 3D atau media lainnya dapat ditampilkan. Vuforia dapat mendeteksi beberapa jenis target atau marker, salah satunya image target. Disini yang dijadikan marker atau penanda adalah sebuah gambar. Proses pelacakan dari marker adalah SDK Vuforia akan mendeteksi dan melacak fitur yang secara alami ditemukan pada gambar itu sendiri dengan membandingkan fitur alami dengan database yang sebelumnya sudah didefinisikan. (Vuforia Developer Library, 2017).



Gambar 2.3 Contoh Marker

2.2.5 Fungsi *Distance*

Fungsi *distance* merupakan salah satu fungsi yang ada di *scripting* unity 3D. Fungsi ini bertujuan untuk mencari jarak antara satu objek dengan objek yang lainnya. Fungsi ini dapat digunakan untuk objek 2D maupun 3D. Untuk objek 3D terkait dengan *Vector3*. Maksudnya adalah objek 3D memiliki 3 buah sumbu yaitu x, y, z. Sehingga fungsi *distance* yang digunakan pada objek 3D adalah *Vector3.Distance*. Fungsi *distance* memiliki 2 nilai parameter *input* yaitu nilai posisi masing-masing dari dua buah objek yang akan dicari jaraknya. Sedangkan untuk mencari jarak antara 3 buah objek menggunakan fungsi *distance* dapat dilakukan dengan cara salah satu objek dijadikan patokan dari dua objek lainnya. Sehingga contoh scriptnya seperti berikut :

```
jarak1 = Vector3.Distance(objekA.position, objekB.position);
jarak2 = Vector3.Distance(objekA.position, objekC.position);
```

(Unity3D Documentation Script Reference, 2017).